高效的可撤销 SM9 标识签名算法 *

张博鑫 [1]，耿生玲 [2]，秦宝东 [1†]

(1. 西安邮电大学 网络空间安全学院，西安 710121; 2. 青海师范学院 计算机学院，西宁 810008)

摘 要： SM9-IBS 是中国于 2016 年公布的一种标识签名算法行业标准。标识签名算法虽然降低了系统管理用户公钥的复杂性，但是却存在密钥撤销的难题。此外，SM9 的特殊结构使得已有技术无法完全适用。为此，提出了一种可撤销 SM9 标识签名算法，可快速实现对用户签名权限的撤销和更新操作。该算法引入一棵完全子树，密钥中心借助该树为每个合法用户生成临时签名密钥，只有使用该密钥生成的签名才可以通过签名验证。在安全性方面，该算法在随机预言机模型中，被证明在适应性选择消息和标识攻击模型下满足存在性不可伪造。在效率方面，该方案在密钥更新阶段当系统用户数量较大、被撤销用户数量较少时，密钥中心更新用户签名密钥的时间开销远小于 Boneh 等人的更新技术。

关键词： 标识密码系统；SM9 签名算法；密钥撤销；完全子树

中图分类号： TP309 doi: 10.19734/j.issn.1001-3695.2022.02.0073

Efficient revocable SM9 identity-based signature algorithm

Zhang Boxin[1], Geng Shengling[2], Qin Baodong[1†]

(1. *School of Cyberspace Security*, *Xi'an University of Posts & Telecommunications*, *Xi'an* 710121, *China*; 2. *School of Computer*, *Qinghai Normal University*, *Xining* 810008, *China*)

Abstract: SM9-IBS is an industry standard for identity-based signature (IBS) algorithms issued by China in 2016. Although the IBS algorithms can be reduce the complexity of management of user keys, they have the problem of key revocation. In addition, the existing technologies are not fully applicable to SM9-IBS due to its special algebraic structure of users' secret keys. Therefore, this paper proposes an efficient revocable SM9 identity-based signature (shorted as CS-SM9-RIBS) algorithm, which can quickly revoke and update the user's signature authority. The algorithm introduces a complete subtree, which is used by the key generation center (KGC) generates temporary signature keys for each legitimate user, so that only the signature generated by this key can pass the signature verification. In terms of security, the new algorithm is proven to be existentially unforgeable under adaptive chosen message and identity attacks in the random oracle model. In terms of efficiency, when the number of users in the system is large and the number of revoked users is small in the key update stage, the time cost of the KGC to update the user's signature key is much smaller than Boneh et al. 's update technology.

Key words: identity-based cryptosystem; SM9 signature algorithm; key revocation; complete subtree

0 引言

1984 年，Shamir[1]首次提出了标识密码(Identity-Based Cryptosystem)的概念，包括标识加密算法(Identity-Based Encryption, IBE)和标识签名算法(Identity-Based Signature, IBS)。它可以采用用户的唯一标识，如身份证号、手机号、电子邮件地址等作为用户公钥，由密钥中心根据系统密钥以及用户标识生成相应的用户私钥。与传统公钥密码算法相比，标识密码算法简化了 PKI/CA 应用中的证书分发和公钥交换的过程，降低了密钥和证书管理的复杂性。2000 年，Ohgishi 和 Kasahara[2]发表了一篇使用椭圆曲线对提出的基于身份的密钥共享方案。直到 2001 年，Boneh 和 Franklin[3]以及 Cocks[4]才分别给出首个标识加密算法(Identity-Based Encryption，简称 IBE)的构造。随后，美国，英国和日本开始设计相应的算法，如文献[5~8]。

中国国家密码学管理局于 2008 年正式发布了 SM9[9]商用标识密码算法，并于 2016 年 3 月 28 日发布并实施了 SM9 标识密码算法标准。SM9 系列算法包括密钥交换，签名以及加密三种，其中签名算法与加密算法分别已于 2018 年 11 月[10]和 2021 年 2 月[11]被纳入国际标准。SM9 因其结构简单，功耗低等优点，在密码技术和网络空间安全领域愈来愈受到广泛地应用与发展，如文献[12~16]。

虽然标识密码体系简化了用户公钥管理的环节，但是包括 SM9 在内的标识密码算法并没有提供用户密钥撤销机制，一旦用户私钥泄露，用户标识就需作废无法继续使用，否则攻击者就会使用泄露的密钥访问用户所有的密文或者代表用户签署任何文献。针对标识密码系统的密钥泄露问题，2001 年，Boneh 等人[3]提出利用身份标识级联时间戳作为用户临时身份，例如 $ID\|2022.01$，密钥中心定期将相应的密钥发送给未被撤销的用户。该方法具有通用性，适用于任意标识密码系统，但是密钥中心更新密钥的复杂度与系统用户数量线性增长，为 $O(N-R)$，其中 N 是系统用户数量，R 是被撤销用户数量。此外，该方法需要安全信道传输用户的密钥。2008 年，Boldyreva 等人[17]提出第一个高效的可撤销 IBE 方案。该方案为每个用户预分配一组长期密钥并利用完全子树覆盖技术实现用户临时密钥的安全更新，从而使得密钥更新的复杂度

收稿日期： 2022-02-18；**修回日期：** 2022-03-30 **基金项目：** 国家自然科学基金资助项目(61872292)；青海省重点研发计划资助项目(2020-SF-139)；青海省基础研究计划资助项目(2020-ZJ-701)

作者简介： 张博鑫(1996-)，男，陕西咸阳人，硕士研究生，主要研究方向为公钥密码学；耿生玲(1970-)，女(藏族)，青海都兰人，教授，副院长，博士，主要研究方向为物联网、大数据、数据挖掘；秦宝东(1982-)，男(通信作者)，江苏徐州人，教授，硕导，博士，主要研究方向为公钥密码学基础理论及应用(qinbaodong@xupt.edu.cn).

从线性降至 $O(R\cdot\log(N/R))$ ，且不需要安全信道传输更新密钥。近年来，一些学者从适应性身份安全性[18]、临时密钥泄露攻击[19]、更新密钥的复杂性[20]、抗量子计算攻击[21,22]、云计算辅助撤销[23~25]等方面，提出了一系列可撤销的标识密码方案。这些方法主要利用完全子树(Complete Subtree，CS)覆盖技术或子集合差分(Subset Difference，SD)技术确定更新节点数量，从而降低更新密钥的复杂性。同时，利用主密钥与用户密钥之间存在的同态性质，计算各个更新节点的更新密钥。然而，SM9"指数倒置"的特殊代数结构，使得构造上述方法无法直接用于 SM9 标识密码系统。2019 年，Ma 等人[26]利用完全子树覆盖技术提出一种构造可撤销标识加密方案的通用方法。与文献[3]中的通用方法相比，该方法将更新密钥的复杂度从 $O(N-R)$ 降至 $O(R\cdot\log(N/R))$ ，但是密文长度从 $O(1)$ 扩展到了 $O(\log N)$，或者依赖基于身份的广播加密。将该方法直接应用到 SM9 上，密文的数量至少为 $O(\log N)$， 效率较低。截止目前，还没有针对 SM9 签名算法的高效密钥撤销机制的研究。

解决思路：针对标识加密算法，Ma 等人[26]提出的更新密钥的主要思想是：通过引入一棵完全子树，将用户的身份标识与该树叶子节点一一对应。在更新密钥时，利用完全子树覆盖技术确定仅覆盖未被撤销用户的最小节点集合，再利用时间戳级联更新节点信息作为更新标识，计算出更新密钥集合。在加密时，由于发送者需要将时间戳与接收者身份标识对应的叶子节点到根节点路径上的所有节点信息级联作为临时身份标识用于加密消息，从而使得密文数量至少为 $O(\log N)$ 。受此思想的启发，针对 SM9 标识签名算法，可以采用同样的更新密钥的策略并利用长期密钥和更新密钥生成消息的两份签名。但是，用户在验证签名的合法性时，除了利用签名者的身份标识进行一次验签外，还要利用至少 $O(\log N)$ 个临时标识进行验签，从而使得签名验证算法的效率较低。为了解决该问题，本文在签名时，将使用的更新密钥的节点信息嵌入到签名中，用户首先验证嵌入的节点信息是否在用户的路径上，再使用该节点信息(作为临时标识)验证签名的合法性，从而使得验签次数降至 2 次。

本文工作：本文提出了一种可撤销 SM9 标识签名算法(简称 CS-SM9-RIBS)，它利用完全子树实现在 SM9 标识签名算法中对用户密钥的撤销和更新机制。在该算法中，密钥中心会根据用户的身份生成一个长期签名密钥，根据更新节点集合生成更新密钥并发送给用户，使得只有未被撤销的用户才能合成一个合法的临时签名密钥，用于签名消息。与 Boneh 等人[3]的方案(简称 BF-SM9-RIBS)相比，该方案不需要数据发送者和服务器之间建立一条安全信道，并且更新密钥的数量也大大降低：从 $O(N-R)$ 降至 $O(R\cdot\log(N/R))$ 。通过实验分析表明，当系统用户数量为 $N=2^{13}=8192$ ， 被撤销用户数量 $R\in[0,100]$ 时，BF-SM9-RIBS 方案密钥更新的时间约为 45 秒，而 CS-SM9-RIBS 方案所需要时间在 15 秒以下。

1　基础知识

1.1　符号说明

在本文中，" $A\|B$ "表示将两个比特串 A 和 B 进行级联。若 S 是一个算法，则 $s\leftarrow S$ 表示执行算法 S 输出的结果为 s；若 S 是一个集合，则 $s\leftarrow S$ 表示从集合 S 中以均匀随机分布选取一个元素 s。

在 SM9 算法中，

a) H_v 为输入为任意比特串，输出为 v 比特的密码杂凑函数。

b) $\mathrm{H2RF}(H_v,Z,p)$ 为输入为密码杂凑函数 H_v ，比特串 Z 和整数 p，输出为整数 $h\in[1,p-1]$ 的密码函数。

1.2　双线性映射

令 $\mathcal{P}(1^\lambda)$ 表示一个双线性对群生成算法，输入 1^λ 为安全参数，输出为 $(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h)$ ，其中 $\mathbb{G}_1$ 、 $\mathbb{G}_2$ 和 $\mathbb{G}_T$ 是阶为素数 p 的循环群，映射 $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ 为双线性对， g 和 h 分别是 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 的生成元。一个双线性映射应该满足以下性质：

a) 双线性性：对于任意的 $\alpha,\beta\in\mathbb{Z}_p$ ， 有 $e(g^\alpha,h^\beta)=e(g,h)^{\alpha\beta}$ ；

b) 非退化性： $\exists g_1\in\mathbb{G}_1,g_2\in\mathbb{G}_2$ 满足 $e(g_1,g_2)\neq 1_{\mathbb{G}_T}$ ；

c) 可计算性： 对于任意的 g 和 h ，存在有效多项式时间算法计算 $e(g,h)$ 。

1.3　困难问题假设

本节主要介绍文献[27]提出的 SM9 签名算法满足适应性选择消息和标识攻击下的存在性不可伪造攻击安全性(Existentially Unforgeable against adaptive Chosen Message and Identity Attacks，简称 EU-CMIA 安全性)依赖的 q-SDH 问题(q-strong Diffie-Hellman)。令 $e:\mathbb{G}_1\times\mathbb{G}_2\to\mathbb{G}_T$ 为一个双线性映射， g 和 h 分别是 $\mathbb{G}_1$ 和 $\mathbb{G}_2$ 的生成元，群 $\mathbb{G}_1$ 、 $\mathbb{G}_2$ 和 $\mathbb{G}_T$ 的阶为 p 。则基于双线性群的 q-SDH 问题的定义如下：

定义 1　q-SDH 问题。已知 $q+2$ 个群元素 $(g,h,h^a,h^{a^2},\ldots,h^{a^q})\in\mathbb{G}_1\times\mathbb{G}_2^{q+1}$ ，其中 a 未知，找到一个二元组 $(c,g^{1/(c+a)})$ ，其中 $c\in\mathbb{Z}_p$ 。

若 q-SDH 问题在多项式时间内可解的概率是可忽略的，则称 q-SDH 假设成立。

1.4　SM9-IBS 标识签名算法

SM9 标识签名算法(简称 SM9-IBS 算法)，包括 4 个(概率)多项式时间算法：系统参数生成算法、用户密钥提取算法、签名生成算法和验证算法。

下面本文简要回顾 SM9-IBS 算法。

a) 系统参数 $\mathrm{SM9.Setup}(1^\lambda)$ ：输入安全参数 1^λ ，密钥中心首先运行算法 $\mathcal{P}(1^\lambda)$ 生成一个双线性对配对群 $\mathcal{G}=(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h)$ ；然后，选取随机数 $s\in\mathbb{Z}_p$ ，计算 $P_{pub-s}=g^s$ 和 $u=e(g,h)^s$ 。则系统的主公钥和主私钥分别为

$$mpk=(\mathcal{G},P_{pub-s},u,H(id),hid) \tag{1}$$

$$msk=s \tag{2}$$

其中 $H(id)$ 是密码函数 $\mathrm{H2RF}(H_v,id,p)$ ，$hid=3$ 。假设系统的主公钥作为其他算法的默认输入，而省略不写。

b) 用户密钥 $\mathrm{SM9.Extract}(msk,id)$ ： 对于身份标识为 id 的用户，密钥生成中心为用户计算签名私钥为

$$ds_{id}=g^{\frac{s}{H_1(id\|hid,N)+s}} \tag{3}$$

则签名者公钥为 id ， 私钥为 ds_{id} 。

c) 签名 $\mathrm{SM9.Sign}(ds_{id},M)$ ： 身份标识为 id 的签名者对消息 M 进行签名，签名过程如下：

(a) 计算 $w=u^r$ ， 其中 r 为随机数且满足 $r\in\mathbb{Z}_p$ ；

(b) 计算 $h=\mathrm{H2RF}(H_v,M\|w,p)$ ；

(c) 计算 $l=(r-h)\bmod p$ ， 若 $l=0$ ， 则返回第二步；

(d) 计算 $S=ds_{id}^l$ 。

则签名者对消息的签名结果为 $\sigma=(h,S)$ 。

d) 验证 $\mathrm{SM9.Verify}(id,M',\sigma')$ ：验证者在收到签名者的身份标识 id ，消息 M' 和签名结果 $\sigma'=(h',S')$ 后，执行以下步骤进行签名验证。

(a) 计算 $t=u^{h'}$ ；

(b) 计算 $P=h^{H(id)}\cdot P_{pub-s}$ ；

(c) 计算 $w'=\alpha\cdot t$ ， 其中 $\alpha=e(S',P)$ ；

(d) 计算 $h_2=\mathrm{H2RF}(H_v,M'\|w',p)$ ；

(e) 若 $h_2=h'$ ， 则验签通过； 否则验签失败。

注释 1：为了提高签名和验证算法的效率，可以将原始 SM9-IBS 算法中计算的固定值 $u=e(g,P_{pub-s})$ 放到系统公钥中，从而使得签名和验证算法都减少一次双线性配对运算，而系统公钥将增加一个群 $\mathbb{G}_T$ 上的元素。

标识签名算法的 EU-CMIA 安全模型：在标识签名算法

中，适应性选择消息和标识攻击下的存在性不可伪造 EU-CMIA 安全模型的定义与传统签名算法的安全模型定义类似，除了平凡的询问外，允许攻击者选择任意标识并获取相应的签名密钥，以及任意消息并获取签名结果，具体参见文献[27]。在该模型下，有以下结论：

定理 1　SM9-IBS 的安全性[27]。在随机预言机模型下，如果 q-SDH 问题是困难的，则 SM9-IBS 满足 EU-CMIA 安全性。

2　CS-SM9-RIBS 标识签名算法

2.1　RIBS 的定义

图 1 给出了可撤销标识数字签名算法的系统模型，它包含以下四个实体：

a) 密钥中心(KGC)：主要负责产生系统主公钥 mpk 和主私钥 msk，为身份标识为 id 的用户生成长期签名密钥 ds_{id}，生成更新密钥集合 uk_t，维护用户撤销列表 RL 及一个与所有用户身份标识 id 对应的身份二叉树 Tree。

b) 数据签名者(DS)：主要负责计算原始消息 M 的数字签名结果 σ。

c) 存储服务器(SS)：主要负责存储用户的签名。

d) 签名验证者(SV)：主要负责对接收到的签名 σ 进行合法性验证。

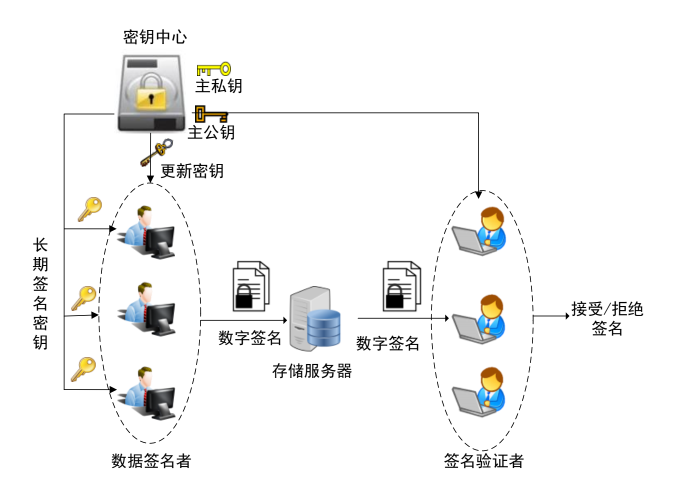


图 1　RIBS 系统模型

Fig. 1　System model of RIBS

定义 2　RIBS 的定义。一个 RIBS 算法包含系统参数生成算法、用户注册算法、未撤销用户的更新节点生成算法、更新密钥生成算法、临时签名密钥生成算法、签名算法、验证算法和用户撤销算法 8 个(概率)多项式时间算法，具体如下：

a) 系统参数 $\mathrm{Setup}(1^\lambda)$：该算法由密钥中心负责执行。它的输入为安全参数 λ，输出为系统的主公钥 mpk 和主私钥 msk。此外，密钥中心维护一个初始化为空集的撤销列表 RL，以及一个与所有用户身份标识 id 对应的满二叉树 Tree。

b) 用户注册 $\mathrm{Regist}(msk,id)$：该算法由密钥生成中心负责执行。它的输入为系统的主私钥和用户的身份标识 id，将用户的身份标识 id 加入到 Tree，并输出用户的长期签名密钥 ds_{id}。

c) 更新节点 $\mathrm{KUNode}(\mathrm{Tree},RL,t)$：该算法由密钥中心执行。它输入身份二叉树 Tree，撤销列表 RL 和时间 t，输出所有需要生成更新密钥的节点集合 $KUNodes$。

d) 更新密钥 $\mathrm{UpdateK}(msk,t,KUNodes)$：该算法由密钥中心执行。输入为系统主私钥 msk，时间 t 以及未撤销结集合 $KUNodes$，输出为更新密钥集合 uk_t，并通过公开信道广播给系统用户。

e) 临时签名密钥 $\mathrm{TempK}(ds_{id},uk_t)$：该算法由数据签名者执行。它输入用户的长期签名密钥 ds_{id} 和更新密钥集合 uk_t。若该用户未被撤销，则算法输出为用户的临时签名密钥 $tsd_{id,t}$（该签名密钥不仅包含了用户的身份 id 和更新的时间 t，同时也包含了对应的更新节点信息 θ）。

f) 签名 $\mathrm{Sign}(tsd_{id,t},M)$：该算法由数据签名者执行。它的输入为原始消息 M，用户临时签名密钥 $tsd_{id,t}$，计算出消息 M 的签名 σ。该签名包含了时间周期和更新节点信息。最后，将消息和签名 (M,σ) 一同发送给存储服务器进行储存。

g) 验证 $\mathrm{Verify}(id,M,\sigma)$：该算法由签名验证者执行。算法的输入为签名者的身份标识 id、消息 M 和签名 σ。输出签名的验证结果。若签名合法，则输出 1，否则输出 0。

h) 用户撤销 $\mathrm{Revoke}(id,t,RL)$：该算法由密钥中心执行。它的输入为被撤销的用户身份标识 id、时间周期 t 和用户撤销列表 RL。密钥中心将 (t,id) 加入撤销列表，返回更新后的撤销列表 RL。

2.2　安全模型

为刻画方案具有密钥撤销机制，本文通过攻击者与挑战者的游戏描述适应性选择消息和标识攻击下的存在性不可伪造攻击(EU-CMIA 安全性)。在该模型中，攻击者可以询问任意用户的长期签名密钥，当某一用户的签名密钥泄露(被攻击者询问)并在 t 时刻被密钥中心撤销时，方案的安全性能够保证在 $t^*\geq t$ 时刻，攻击者无法生成该用户的一个合法签名。

定义 3　RIBS 的 EU-CMIA 安全性。假设 A 是任意一个概率多项式时间攻击者，C 是一个挑战者。定义 RIBS 的安全性游戏 $\mathrm{Exp}^{\mathrm{EU\text{-}CMIA}}_{\mathrm{RIBS,A}}(1^\lambda)$ 如下：

a) 系统建立：挑战者通过运行系统参数生成算法 $\mathrm{Setup}(1^\lambda)$，生成系统主公钥 mpk 和主私钥 msk，并初始化一个空的密钥撤销列表 RL 和一个满二叉树 Tree。挑战者将系统公钥 mpk 发送给攻击者。

b) 询问：攻击者可以适应性地询问进行以下询问，询问的次数为多项式次：

(a) 用户长期签名密钥询问．挑战者初始化一个空集合 L_1。当攻击者询问身份标识为 id 的用户长期签名密钥时，挑战者通过执行用户注册算法 $\mathrm{Regist}(msk,id)$ 生成密钥 ds_{id} 并返回给攻击者。同时，挑战者将身份标识 id 添加到集合 L_1 中。

(b) 更新密钥询问．当攻击者询问 t 时刻的更新密钥时，挑战者首先执行更新节点生成算法 $\mathrm{KUNode}(\mathrm{Tree},RL,t)$ 得到 t 时刻的更新节点集合 $KUNodes$；然后执行更新密钥生成算法 $\mathrm{UpdateK}(msk,t,KUNodes)$ 得到 t 时刻的更新密钥集合 uk_t，并将它返回给攻击者。

(c) 签名询问．挑战者初始化一个空集合 L_2。当攻击者询问消息 M 在身份标识为 id 和时刻 t 下的签名时，挑战者先利用该用户 id 的长期签名密钥 ds_{id} 和 t 时刻的更新密钥集合 uk_t，通过临时签名密钥生成算法生成该用户的临时签名密钥 $tsd_{id,t}$；然后，利用签名算法 $\mathrm{Sign}(tsd_{id,t},M)$ 生成消息 M 的签名 σ，并将该签名返回给攻击者。同时，挑战者将元素 (id,t,M) 添加到集合 L_2 中。

(d) 用户撤销询问．当攻击者询问 t 时刻的标识为 id 的用户撤销时，挑战者将 (t,id) 添加到用户撤销列表 RL 中。

c) 伪造．攻击者返回一个伪造的签名信息 (M^*,σ^*) 以及身份标识 id^* 和时间 t^*，并满足以下限制条件

(a) 若用户在 t^* 或之前时刻未撤销，即不存在 $t\leq t^*$，使得 $(t,id^*)\in RL$，则攻击者不能访问过身份标识 id^* 的长期签名密钥，即 $id^*\notin L_1$ 并且 $(id^*,t^*,M^*)\notin L_2$；

(b) 若用户在 t^* 或之前时刻已撤销，即存在 $t\leq t^*$，使得 $(t,id^*)\in RL$，则攻击者可以询问身份标识 id^* 的长期签名密钥，即 $id^*\in L_1$。

d) 输出．若攻击者伪造的签名信息满足限制条件，并能

通过签名验证算法，则表示攻击者伪造成功，挑战者返回 1；否则返回 0。

在上述游戏中，A 成功的概率定义为

$$\Pr[\text{A成功}]=\Pr[\text{Exp}_{\text{RIBS},A}^{\text{EU-CMA}}(1^{\lambda})=1] \tag{4}$$

如果 A 成功的概率(关于安全参数 1^{λ})是可忽略的，则称 RIBS 算法是 EU-CMIA 安全的。

2.3　算法设计

本节利用完全子树覆盖技术，设计一种可撤销 SM9 标识签名算法(记作 CS-SM9-RIBS)。具体构造如下：

a) 系统参数 $\text{Setup}(1^{\lambda})$：在输入安全参数 1^{λ} 后，生成系统参数的过程如下：

(a) 密钥中心运行算法 $\mathcal{P}(1^{\lambda})$ 生成一个双线性配对群 $\mathcal{G}=(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h)$；密钥中心选择一个随机元素 $s\in\mathbb{Z}_p$，并计算

$$P_{pub-s}=h^s \text{ 和 } u=e(g,P_{pub-s}) \tag{5}$$

令 $H(id)=\text{H2RF}(H_v,id,p)$、$hid=3$。假设每个用户的身份标识长度为 n 比特，则密钥中心初始化一棵高度为 $n+1$ 的满二叉树 Tree 以及一个空的用户撤销列表 RL。

(b) 输出系统的主公钥 mpk 和主私钥 msk：

$$mpk=(\mathcal{G},P_{pub-s},u,H(\cdot),hid) \tag{6}$$

$$msk=s \tag{7}$$

b) 用户注册 $\text{Register}(msk,id)$：密钥中心在收到身份标识为 id 的用户注册请求后，为用户注册生成一个长期签名密钥的过程如下：

(a) 密钥中心使用 SM9 的密钥提取算法为标识为 id 的用户生成签名密钥 ds_{id}，即：

$$ds_{id}\leftarrow\text{SM9.Extract}(msk,id) \tag{8}$$

(b) 密钥中心选择一个与用户身份标识匹配的叶子节点，将 id 添加到 Tree 上；

(c) 将用户长期签名密钥 ds_{id} 发送给用户。

c) 更新节点 $\text{KUNode}(\text{Tree},RL,t)$：在输入身份二叉树 Tree，撤销列表 RL 和时间 t 后，令 θ 为 Tree 的非叶子节点，θ_l 和 θ_r 分别为 θ 的左子节点和右子节点，$\text{Path}(\theta_{id})$ 为根节点到叶子节点 θ_{id} 路径上所有节点的集合，未撤销节点生成过程如下：

(a) 令集合 $X,Y=\varnothing$；

(b) 对 $\forall(t',id)\in RL$ 且 $t'\le t$，将 $\text{Path}(\theta_{id})$ 中的每一个节点添加到集合 X 中。

(c) 对 $\forall\theta\in X$：

如果 $\theta_l\notin X$，则将 θ_l 添加到集合 Y 中；

如果 $\theta_r\notin X$，则将 θ_r 添加到集合 Y 中。

(d) 如果 $Y=\varnothing$，那么将根节点 root 添加到集合 Y 中；

(e) 令 $KUNodes=Y$，并输出 $KUNodes$。

图 2 和图 3 给出了更新节点选择算法的两个具体示例。在该示例中，二叉树的高度为 4(用户的身份标识长度为 3 比特)，蓝色节点表示当前 $KUNodes$ 所包含的更新节点，虚线节点表示被撤销的用户节点。在图 2 中，无用户被撤销，$KUNodes$ 仅包含根节点，即 $\{root\}$；在图 3 中，标识为 $id=3$ 的用户被撤销，$KUNodes=\{00,010,1\}$。

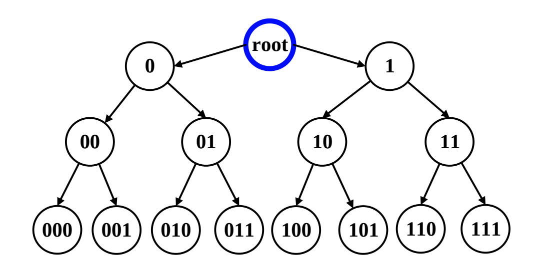


图 2　无用户被撤销

Fig. 2　No user was revoked

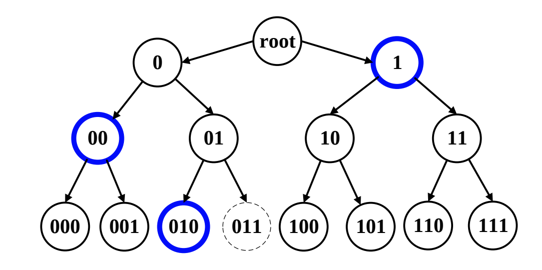


图 3　用户 $id=3$ 被撤销

Fig. 3　The user identified as $id=3$ was revoked

d) 更新密钥 $\text{UpdateK}(msk,t,KUNodes)$：在输入系统密钥 msk，时间 t 以及更新节点集合 $KUNodes$ 后，密钥中心首先计算由时间和更新节点组成的更新标识集合 $UID=\{t\|\theta\}_{\theta\in KUNodes}$。对于集合 UID 中的每个标识 $uid=t\|\theta$，调用 SM9 的密钥提取算法，生成相应的标识密钥，即

$$ds_{t\|\theta}\leftarrow\text{SM9.Extract}(msk,uid) \tag{9}$$

令 $uk_{t,\theta}=ds_{t\|\theta}$。则更新密钥集合为

$$uk_t=\{uk_{t,\theta}\}_{\theta\in KUNodes} \tag{10}$$

密钥中心通过公开信道广播给系统用户。

e)临时签名密钥 $\text{TempK}(ds_{id},uk_t)$：在输入用户的长期签名密钥 ds_{id} 和更新密钥集合 uk_t 后，用户执行以下步骤：

(a) 若该用户未被撤销，则用户路径上的节点集合 $\text{Path}(\theta_{id})$ 与更新节点集合 $KUNodes$ 一定存在一个且唯一的公共节点 θ。用户找到该节点对应的更新密钥 $uk_{t,\theta}$，并将临时签名密钥定义为 $tsd_{id,t}=ds_{id}\|uk_{t,\theta}\|\theta$；若该用户已被撤销，则不存在公共的更新节点 θ，用户将临时签名密钥定义为终止符，即 $tsd_{id,t}=\perp$；

(b) 返回临时签名密钥 $tsd_{id,t}$。

f) 签名 $\text{Sign}(tsd_{id,t},M)$：在输入明文消息 M，用户的临时签名密钥 $tsd_{id,t}$ 后，该算法执行过程如下：

(a) 若临时签名密钥 $tsd_{id,t}=\perp$，则用户终止算法并返回 $\perp$；否则执行步骤(b)；

(b) 根据临时签名密钥恢复出用户的长期签名密钥 ds_{id}，更新密钥 $uk_{t,\theta}$ 以及更新节点信息 θ，并令 $\bar{M}=M\|t\|\theta$ (假设不同的消息、时间和更新节点级联的结果不同)。

(c) 以 ds_{id} 作为签名密钥，执行 SM9 签名算法，生成消息 $\bar{M}$ 的第一部分签名 $\sigma_1=(h_1,S_1)$，即

$$(h_1,S_1)\leftarrow\text{SM9.Sign}(ds_{id},\bar{M}) \tag{11}$$

(d) 以 $uk_{t,\theta}$ 作为签名密钥，再次执行 SM9 签名算法，生成消息 $\bar{M}$ 的第二部分签名 $\sigma_2=(h_2,S_2)$，即

$$(h_2,S_2)\leftarrow\text{SM9.Sign}(uk_{t,\theta},\bar{M}) \tag{12}$$

(e) 令 $\sigma=(\sigma_1,\sigma_2,t\|\theta)$ 为最终签名发送给存储服务器进行存储。

g) 签名验证 $\text{Verify}(id,M,\sigma)$：在输入签名者的身份标识 id、消息 M 和签名 σ 后，签名验证过程如下：

(a) 签名验证者收到签名 $\sigma=(\sigma_1,\sigma_2,t\|\theta)$ 之后，将其拆分为 σ_1，σ_2 和 $t\|\theta$ 三部分。验证者首先判断节点 θ 是否在路径 $\text{Path}(\theta_{id})$ 上。如果不是，则算法终止并返回 0；否则，执行步骤(b)；

(b) 验证者计算 $\bar{M}=M\|t\|\theta$，使用签名者的身份标识 id 利用 SM9 签名验证算法对消息 $\bar{M}$ 和签名 σ_1 进行验证，验证结果记为 b_1，即：

$$b_1\leftarrow\text{SM9.Verify}(id,\bar{M},\sigma_1) \tag{13}$$

如果 $b_1=0$，则算法终止并返回 0；否则，执行步骤(c)；

(c) 接下来验证者使用更新标识 $t\|\theta$ 利用 SM9 签名验证算法对消息 $\bar{M}$ 和签名 σ_2 进行验证，验证结果记为 b_2，即：

$$b_2\leftarrow\text{SM9.Verify}(t\|\theta,\bar{M},\sigma_2) \tag{14}$$

如果 $b_2=0$，则算法终止并返回 0；否则，执行步骤(d)；

(d) 上述验证均通过，算法返回 1。

h) 用户撤销 $\text{Revoke}(id,t,RL)$：在输入被撤销用户身份标识 id、时间周期 t 和撤销列表 RL 后，若存在元素 $(t',id)\in RL$ 且 $t'\le t$，则直接返回；否则，将 (t,id) 加入撤销列表 RL，返回 $RL\cup(t,id)$。

正确性：假设 $\sigma=(\sigma_1,\sigma_2,t\|\theta)$ 是利用签名算法 $\text{Sign}(M,tsd_{id,t})$ 计算的签名。因为 $\sigma_1=(h_1,S_1)$ 是利用标识为 id 的 SM9 签名密钥 ds_{id} 对消息 $\bar{M}$ 的签名，根据 SM9 签名算法的正确性，则验证算法在第(b)步输出结果应为 $b_1=1$；

若用户 id 在时刻 t 未被撤销，根据更新节点选择算法，用户能够从更新密钥中获取临时签名密钥 $uk_{t,\theta}$。该密钥对应的身份标识为 $t\|\theta$，并且 θ 属于路径 $\text{Path}(\theta_{id})$ 上的点。因此，步骤(a)的验证通过。由于 $\sigma_2=(h_2,S_2)$ 是利用标识为 $t\|\theta$ 的 SM9 签名密钥 $uk_{t,\theta}$ 对消息 $\bar{M}$ 的签名，根据 SM9 签名算法的正确性，则验证算法在第(c)步输出结果应为 $b_2=1$。

综上，CS-SM9-RIBS 签名算法满足正确性。

3 安全性分析

定理 2 CS-SM9-RIBS 的安全性。假设 A 是任意一个概率多项式时间算法，Π 是一个 CS-SM9-RIBS 签名算法。若 A 能够以概率 ε_Π 成功攻击 Π 的 EU-CMIA 安全性，那么存在另一个概率多项式时间算法 B，以相同的概率成功伪造原始 SM9-IBS 的一个签名。

证明 本文利用算法 A 作为子程序，构造一个仿真算法 B 在 EU-CMIA 模型下，伪造 SM9-IBS 签名算法的签名。仿真算法 B 按照如下方式模拟攻击者 A 在游戏 $\text{Exp}_{\text{RIBS},A}^{\text{EU-CMIA}}(1^\lambda)$ 中的环境。

a) 系统建立：仿真算法 B 首先询问 SM9-IBS 的挑战者，获取 SM9-IBS 的一个挑战主公钥 mpk，包括双线性对配对群 $\mathcal{G}=(e,\mathbb{G}_1,\mathbb{G}_2,\mathbb{G}_T,p,g,h)$，公钥元素 P_{pub-s}、u 和密码函数 $H(id)$ 及其标识 hid。B 将主公钥 mpk 发送给攻击者 A。同时，B 初始化一个空的密钥撤销列表 RL、一棵满二叉树 Tree 以及两个空集合 L_1 和 L_2。

b) 询问：B 通过如下方式回答攻击者的询问：

(a) 用户长期签名密钥询问. 当攻击者 A 询问身份标识为 id 的用户长期签名密钥时，B 将 id 发送给 SM9-IBS 的挑战者，从而获取身份标识为 id 的密钥 ds_{id} 并将它返回给攻击者 A。同时，A 将身 (id,ds_{id}) 添加到集合 L_1 中。

(b) 更新密钥询问. 当 A 询问 t 时刻的更新密钥时，B 首先根据算法 $\text{KUNode}(\text{Tree},RL,t)$ 计算 t 时刻的更新节点集合 $KUNodes$；然后确定更新节点标识集合 $UID=\{t\|\theta\}_{\theta\in KUNodes}$。对于集合 UID 中的每个标识 $t\|\theta$，B 询问 SM9-IBS 挑战者，获取该标识的密钥 $ds_{t\|\theta}$。B 将密钥集合 $uk_t=\{ds_{t\|\theta}\}_{\theta\in KUNodes}$ 发送给 A。同时，B 将所有 $(t\|\theta,ds_{t\|\theta})$ 添加到集合 L_1 中。

(c) 签名询问. 当 A 询问消息 M 在身份标识为 id 和时刻 t 下的签名时，仿真算法 B 首先查找集合 L_1 中是否存在以 $t\|\theta$ 为标识的密钥 $ds_{t\|\theta}$，其中 θ 必须在路径 $\text{Path}(\theta_{id})$ 上(注：本文假设 A 总是可以先询问 t 时刻的更新密钥)。若不存在，则 B 直接返回 $\perp$ 给 A(表示在该时刻 id 已被撤销)。否则，B 利用密钥 $ds_{t\|\theta}$ 对消息 $\bar{M}=M\|t\|\theta$ 进行签名，得到第二部分签名 $\sigma_2=(h_2,S_2)$。接下来，B 查找集合 L_1 中是否存在以 id 为标识的密钥 ds_{id}。若存在，则 B 利用密钥 ds_{id} 对消息 $\bar{M}=M\|t\|\theta$ 进行签名，得到第一部分签名 $\sigma_1=(h_1,S_1)$；若不存在，B 将消息 $\bar{M}$ 以及(身份)标识 $t\|\theta$ 发送给 SM9-IBS 的挑战者，获得消息 $\bar{M}$ 在(身份)标识 $t\|\theta$ 密钥下的签名 $\sigma_2=(h_2,S_2)$。最后，B 将签名 $\sigma=(\sigma_1,\sigma_2,t\|\theta)$ 发送给 A。同时，B 将信息 (id,t,M) 添加到集合 L_2 中。

(d) 用户撤销询问. 当 A 在 t 时刻以身份标识 id 询问撤销列表时，的撤销的用户撤销时，B 首先查找集合 RL 是否存在元素 (t',id)，其中 $t'\le t$。若存在，则返回原撤销列表 RL；否则，返回 $RL\cup(t,id)$。

c) 伪造. 当攻击者 A 结束上述询问后，输出一个伪造的签名信息 (M^*,σ^*) 以及挑战身份标识 id^* 和时间 t^*。

d) 输出. 若 (M^*,σ^*) 是一个合法的 CS-SM9-RIBS 签名，其中 $\sigma^*=(\sigma_1^*,\sigma_2^*,t^*\|\theta^*)$，则该签名必须通过签名验证算法。特别地，$\theta^*$ 必须是用户路径。则 B 定义消息 $\overline{M^*}=M^*\|t^*\|\theta^*$，并按如下方式返回消息 $\overline{M^*}$ 的一个伪造 SM9-IBS 签名 $\overline{\sigma^*}$：

(a) 若 $(id^*,\cdot)\notin L_1$，表明 A 从未询问过身份标识为 id^* 的密钥，则 B 输出伪造签名 $\overline{\sigma^*}=\sigma_1^*$；

(b)若存在 $(id^*,\cdot)\in L_1$，则表明 A 询问过身份标识 id^* 的密钥。根据游戏规则，该用户必须在 t^* 之前的某时刻 t 被撤销，即 $(t,id^*)\in RL$。根据更新节点生成算法，在 t^* 时刻，算法 $\text{KUNode}(\text{Tree},RL,t^*)$ 输出的更新节点集合 $KUNodes$ 肯定不包含挑战用户的路径 $\text{Path}(\theta_{id^*})$ 上的节点 θ^*。因此，在整个游戏过程中，B 从未询问过标识为 $t^*\|\theta^*$ 的密钥及其签名。此时，B 输出伪造签名 $\overline{\sigma^*}=\sigma_2^*$。

通过上述分析，B 能够完美地仿真 A 在 CS-SM9-RIBS 中的游戏环境。同时，若攻击者能够成功伪造一个 CS-SM9-RIBS 签名，则仿真算法以相同的概率伪造一个 SM9-IBS 签名。

定理 2 证毕！

4 性能分析

目前，除了直接应用 Boneh-Franklin[3]通用密钥撤销思想可以构造针对 SM9-IBS 的密钥撤销算法外，还没有公开发表的针对 SM9-IBS 的密钥撤销算法。但是存在一些优秀的针对其他标识身份签名的密钥撤销算法，如[19,28~30]。为此，本节首先从理论和实验两个角度分析和比较与 SM9-IBS 相关的密钥撤销算法的性能，包括本文提出的基于完全子树密钥撤销技术的 CS-SM9-RIBS 算法、原始 SM9-IBS 签名算法[9]以及应用 Boneh-Franklin[3]通用密钥撤销技术的 BF-SM9-RIBS 算法。再从理论上分析本文提出的 CS-SM9-RIBS 算法与国际上其他可撤销标识签名算法的优势。

表 1 从参数大小、算法时间及安全性等理论角度对上述与 SM9-IBS 相关的三个算法的性能进行总结与比较。在这些算法中，本文选择使用对称双线性配对，即 $e:\mathbb{G}\times\mathbb{G}\to\mathbb{G}_T$，其中群的阶为素数 p。在表中，N 和 R 分别表示系统用户数量和被撤销用户的数量；“E”和“P”分别表示一次群元素的指数运算和配对运算。在比较各算法所需运算操作次数时，忽略了除指数运算和配对运算之外的其他操作。表中符号“-”表示该项不存在，“0”表示该项几乎不需要任何操作。

在 BF-SM9-RIBS 算法中，用户只需要保存每个周期的临时密钥，而无长期密钥。每个周期的临时签名密钥由密钥中心临时生成的更新密钥构成。更新用户签名密钥的策略是将用户的身份标识 id 与时间周期 t 级联的结果 $id\|t$ 表示当前时刻签名的签名公钥并将相应的私钥通过安全信道发送给相应的为撤销用户。因此，更新用户密钥的复杂度为 $O(N-R)$，与系统用户数量线性增长。对于本文提出的算法，采用完全子树技术更新用户密钥，在平均情况下，更新密钥的数量仅为 $O(R\cdot\log(N/R))$ 且可以在公开信道上进行传输。

从表 1 中的比较可以看出，BF-SM9-RIBS 和 CS-SM9-RIBS 两种算法都支持用户密钥撤销功能，同时保留了原始 SM9-IBS 算法的系统参数。在 BF-SM9-RIBS 算法中，用户不需要存储长期签名密钥并且临时签名密钥仅为 1 个群元素。在算法 CS-SM9-RIBS 中，用户需要存储 1 个长期签名密钥。尽管临时签名密钥包含 2 个群元素，但是它包括了用户的长

期签名密钥，不需要额外空间存储该群元素。在签名大小、签名时间和验证时间方面，本文的 CS-SM9-RIBS 算法是原始 SM9-IBS 算法的 2 倍。这是因为，前者为了实现用户密钥的撤销功能，增加了一个额外的签名操作。BF-SM9-RIBS 算法在参数大小和计算效率方面和原始 SM9-IBS 算法几乎一样。但是该算法的密钥更新复杂度非常高且不支持公开信道传播更新密钥。

表 1　性能比较

Tab. 1　Comparison of performance

	SM9-IBS 文献[9]	BF-SM9-RIBS 文献[3]	CS-SM9-RIBS 第 2.3 节
系统公钥大小	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$	$3\|\mathbb{G}\|+\|\mathbb{G}_T\|$
系统私钥大小	$\|p\|$	$\|p\|$	$\|p\|$
长期密钥大小	$\|\mathbb{G}\|$	-	$\|\mathbb{G}\|$
更新密钥大小	-	$O(N-R)\|\mathbb{G}\|$	$O(R\log(N/R))\|\mathbb{G}\|$
临时密钥大小	-	$\|\mathbb{G}\|$	$2\|\mathbb{G}\|$
签名大小	$\|\mathbb{G}\|+\|p\|$	$\|\mathbb{G}\|+\|p\|$	$2\|\mathbb{G}\|+2\|p\|$
长期密钥时间	1E	-	1E
更新密钥时间	-	$O(N-R)$E	$O(R\log(N/R))$E
临时密钥时间	-	0	0
用户撤销时间	-	0	0
签名时间	2E	2E	4E
验证时间	1P+2E	1P+2E	2P+4E
密钥更新信道	-	安全信道	公开信道
密钥撤销功能	不支持	支持	支持

本文在同一环境下对各算法进行了仿真实现，并测试了平均执行时间。编程测试环境为：2.4GHz Inteli5-9300H CPU、16.0 GB 内存、Windows10 家庭中文版操作系统。

本文主采用 JPBC 库对上述三种 SM9 签名算法进行仿真实现，选取的椭圆曲线类型为 Type-F。表 2 列出了主要算法运行一次的平均时间，未考虑用户撤销等几乎不耗时的算法以及仅需要在系统建立时运行一次的系统参数生成算法。测试的算法主要包括生成长期签名密钥的用户注册(或密钥提取)算法、密钥更新算法(包括更新节点选取算法)、临时签名密钥生成算法、签名算法和验证算法。对于更新密钥生成算法，运行时间不仅与系统用户数量相关，而且与更新节点集合的大小相关。而更新节点集合的大小又与被撤销用户的数量及其在二叉树中的位置相关。为了简化分析，表 2 仅考虑系统中有 100 个用户且无用户被撤销的情况。根据理论分析，BF-SM9-RIBS 算法需要生成 100 个更新密钥，而本文算法仅需要生成一个更新密钥。

表 2　平均执行时间比较

Tab. 2　Comparison of average execution times

	SM9-IBS 文献[9]	BF-SM9-RIBS 文献[3]	CS-SM9-RIBS 第 2.3 节
长期密钥	0.006s	-	0.006s
更新密钥	-	0.685s	0.006s
临时密钥	-	0s	0s
签名	0.043s	0.042s	0.088s
验证	0.467s	0.478s	0.852s

从表 2 可以看出，BF-SM9-RIBS 算法生成更新密钥的时间较慢，而本文的密钥更新算法很快。

为了更加清晰的比较 CS-SM9-RIBS 算法与 BF-SM9-RIBS 算法更新用户密钥的性能差异，测试方案设计如下：初始化的系统用户数量 N 上限为 8192(即 Tree 高为 14)。被撤销用户的数量 R 从 0 增加到 100，且随机选取被撤销用户的位置。图 4 展示当 R 取不同值时更新密钥生成算法的时间开销。

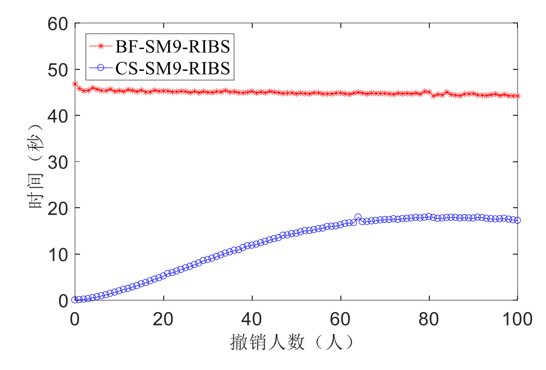


图 4　更新密钥生成算法时间开销对比图

Fig. 4　Comparison of time cost of update key generation algorithm

从图 4 可以看出，当系统用户数量较大，被撤销用户数量较少时，密钥中心利用完全子树技术更新用户签名密钥的时间开销要远远小于 Boneh 和 Franklin 的直接密钥更新技术。

最后，本文从理论上比较 CS-SM9-RIBS 算法与国际上针对其他标识签名的密钥撤销算法[19,28~30]的性能，进一步说明 CS-SM9-IBS 可撤销标识签名算法的优势。在这些算法中，除了文献[30]基于三线性映射群外，其他算法都是基于标准的双线映射群设计的。表 3 将给出具体的比较结果。在表 3 中，除了文献[30]外，其他文献中的符号含义与表 1 中的符号一致。对于文献[30]，$\mathbb{G}_2$ 表示三线性映射中的一个阶为素数 p 的群，而“E”和“P”分别表示三线性映射中的群元素指数运算和配对运算。

表 3　与其他可撤销标识签名算法比较

Tab. 3　Comparisons with other revocable IBS algorithms

	文献[19]	文献[28]	文献[29]	文献[30]	CS-SM9-RIBS
临时签名密钥大小	$3\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$3\|\mathbb{G}\|$	$3\|\mathbb{G}_2\|$	$2\|\mathbb{G}\|$
签名大小	$4\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$4\|\mathbb{G}\|$	$4\|\mathbb{G}_2\|$	$2\|\mathbb{G}\|+2\|p\|$
签名时间	6E	$O(1)$E	3E	6E	4E
验证时间	4P	$O(1)$P	4P+E	4P	2P+4E
更新密钥时间	$O(R\log(N-R))\cdot$E	$O(R\log(N-R))\cdot$E	$O(N-R)\cdot$E	$O(R)\cdot$E	$O(R\log(N/R))\cdot$E

从表 3 可以看出，CS-SM9-RIBS 算法的临时签名密钥和签名大小比其他方案较小。签名时间相较文献[29]多一次指数运算，但是文献[29]的更新密钥时间与系统用户数量线性增长。文献[30]的更新密钥仅与被撤销用户数量相关，但是该方案依赖三线性映射，没有双线性映射的理论基础成熟。总体而言，本文提出的 CS-SM9-RIBS 可撤销标识签名算法的性能优势更明显。

5　结束语

针对 SM9 标识签名算法存在的密钥撤销问题，本文提出了一种可撤销 SM9 标识签名算法。通过一棵完全子树的辅助，可以实现对用户权限的控制，从而实现密钥的撤销和更新机制。通过理论分析和实验表明，该方案相比于 BF-SM9-RIBS 算法在更新密钥方面具有一定的优势。在后续的工作

中，存在以下问题值得进一步研究：该方案相比于 BF-SM9-RIBS 算法在签名和验签阶段多消耗一倍的时间，可进一步研究耗时更少的 SM9 标识签名算法的密钥撤销机制。

参考文献：

[1] Shamir A. Identity-based cryptosystems and signature schemes [C]// Proc of the 4th Advances in Cryptology-Crypto. Berlin: Springer, 1984: 47-53.

[2] Sakai R, Ohgishi K, Kasahara M. Cryptosystems based on pairing [J]. Symposium Cryptographic Information Security, 2000 (43): 26–28.

[3] Boneh D, Franklin M. Identity-based encryption from the Weil pairing [C]// Proc of the 21th Advances in Cryptology-Crypto. Berlin: Springer, 2001: 213-229.

[4] Cocks C. An identity-based encryption scheme based on quadratic residues [C]// Proc of the IMA international conference on cryptography and coding. Berlin: Springer, 2001: 360-363.

[5] Hofheinz D, Jia D, Pan J. Identity-based encryption tightly secure under chosen-ciphertext attacks [C]// Proc of the 29th Advances in Cryptology–Asiacrypt. Berlin: Springer, 2018: 190-220.

[6] Langrehr R, Pan J. Tightly secure hierarchical identity-based Encryption [J]. Journal of Cryptology, 2020, 33 (4): 1787-1821.

[7] Sahai A, Waters B. Fuzzy identity-based encryption [C]// Proc of the 11th Advances in Cryptology–Eurocrypt. Berlin: Springer, 2005: 457-473.

[8] Waters B. Efficient identity-based encryption without random oracles [C]// Proc of the 11th Advances in Cryptology–Eurocrypt. Berlin: Springer, 2005: 114-127.

[9] Cheng Zhaohui. The SM9 cryptographic schemes [J]. Cryptology ePrint Archive, 2017.

[10] ISO/IEC 14888-3: 2018, Information technology security techniques: digital signatures with appendix Part 3: Discrete logarithm based mechanisms [S]. 2018.

[11] ISO/IEC 18033-5: 2015/AMD 1: 2021 Information technology security techniques: encryption algorithms Part 5: Identity-based ciphers Amendment 1: SM9 mechanism [S]. 2021.

[12] Ji Honghan, Zhang Hongjie, Shao Lisong, *et al.* An efficient attribute-based encryption scheme based on SM9 encryption algorithm for dispatching and control cloud [J]. Connection Science, 2021, 33 (4): 1094-1115.

[13] Mu Yongheng, Xyu Haixia, Li Peili, *et al.* Secure two-party SM9 signing [J]. Science China Information Sciences, 2020, 63 (8): 1-3.

[14] 张昱，孙光民，李煜．基于 SM9 算法的移动互联网身份认证方案研究 [J]．信息网络安全, 2021, 21 (4): 1-9. (Zhang Yyu, Sun Guangmin, Li Yyu. Research on Mobile Internet Authentication Scheme Based on SM9 Algorithm [J]. Netinfo Security, 2021, 21 (4): 1-9.)

[15] 姚英英，常晓林，甄平．基于区块链的去中心化身份认证及密钥管理方案 [J]．网络空间安全, 2019, 10 (6): 33-39. (Yao Yingying, Chang Xiaolin, Zhen Ping. Decentralized identity authentication and key management scheme based on blockchain [J]. Cyberspace Security, 2019, 10 (6): 33-39.)

[16] 马晓婷，马文平，刘小雪．基于区块链技术的跨域认证方案 [J]．电子学报, 2018, 46 (11): 2571. (MA Xiaoting, MA Wenping, LIU Xiaoxue. A cross domain authentication scheme based on blockchain technology [J]. Acta Electonica Sinica, 2018, 46 (11): 2571.)

[17] Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation [C]// Proc of the 15th ACM conference on Computer and communications security, New York: ACM Press, 2008: 417-426.

[18] Libert B, Vergnaud D. Adaptive-ID secure revocable identity-based encryption [C]// Proc of the 9th Cryptographers'Track at the RSA Conference. Berlin: Springer, 2009: 1-15.

[19] Seo J H, Emura K. Revocable identity-based cryptosystem revisited: security models and constructions [J]. IEEE Trans on Information Forensics and Security, 2014, 9 (7): 1193-1205.

[20] Lee K, Lee D H, Park J W. Efficient revocable identity-based encryption via subset difference methods [J]. Designs, Codes and Cryptography, 2017, 85 (1): 39-76.

[21] Katsumata S, Matsuda T, Takayasu A. Lattice-based revocable (hierarchical) IBE with decryption key exposure resistance [J]. Theoretical Computer Science, 2020, 809: 103-136.

[22] Takayasu A, Watanabe Y. Lattice-based revocable identity-based encryption with bounded decryption key exposure resistance [C]// Proc of Australasian Conference on Information Security and Privacy. Berlin: Springer, 2017: 184-204.

[23] Li Jin, Li Jingwei, Chen Xiaofeng, *et al.* Identity-based encryption with outsourced revocation in cloud computing [J]. IEEE Trans on Computers, 2013, 64 (2): 425-437.

[24] Qin Baodong, Deng R H, Li Yingjiu, *et al.* Server-aided revocable identity-based encryption [C]// European Symposium on Research in Computer Security. Berlin: Springer 2015: 286-304.

[25] Qin Baodong, Liu Ximeng, Wei Zhuo, *et al.* Space efficient revocable IBE for mobile devices in cloud computing [J]. Science China Information Sciences, 2020, 63 (3): 1.

[26] Ma Xuecheng, Lin Dongdai. Generic constructions of revocable identity-based encryption [C]// Proc of the International Conference on Information Security and Cryptology. Berlin: Springer, 2019: 381-396.

[27] 赖建昌，黄欣沂，何德彪，等．国密 SM9 数字签名和密钥封装算法的安全性分析 [J]．中国科学：信息科学, 2021, 51 (11): 1900-1913. (Lai Jianchang, Huang Xinyi, He Debiao, *et al.* Security analysis of SM9 digital signature and key encapsulation [J]. SCIENTIA SINICA Informationis, 2021, 51 (11): 1900-1913.)

[28] Wei Jianghong, Liu Wenfen, Hu Xuexia. Forward-secure identity-based signature with efficient revocation [J]. International Journal of Computer Mathematics, 2017, 94 (7): 1390-1411.

[29] Yang Xiaodong, Ma Tingchun, Yang Ping, *et al.* Security analysis of a revocable and strongly unforgeable identity-based signature scheme [J]. Information Technology and Control, 2018, 47 (3): 575-587.

[30] Zhao Jing, Wei Bin, Su Yang. Communication-efficient revocable identity-based signature from multilinear maps [J]. Journal of Ambient Intelligence and Humanized Computing, 2019, 10 (1): 187-198.